EDUQ
8879786

INFORMATION DOCUMENT

INTRODUCTION TO COMPUTER SCIENCE ICS 611-434, 611-534 EVALUATION

Québec

Niveau(x) d'enseignement: 4 Numéro EDUQ : 6870786
Centre d'intérêt: 046

Introduction to computer science : ICS 611-434, 611-534, evaluation / Direction générale des programmes, Direction de la formation générale ; [translation by Howard Scott]. -- [Québec] : Ministère de l'éducation, c1986.

51 p. ; 26 cm. -- (Information document / Ministère de l'éducation du Québec, Direction générale des programmes)

Code: 16-0061-04A ; 0889-458.

ISBN 2-550-14438-4

Vedette(s)-secundaire(s): Québec (Province) Ministère de l'éducation. Direction de la formation générale

Descripteur(s): Initiation à l'informatique * Méthode d'évaluation * Méthode d'enseignement * Guide pédagogique * Secondaire (Ordre d'enseignement)

Divers documents traitant du programme d'initiation à l'informatique ont été publiés. Chacun d'eux veut aider à améliorer les relations entre l'enseignant et les élèves, en fournissant aux enseignants des suggestions sur des méthodes particulières d'enseignement et des résolutions de problème. Le document traite de l'évaluation. Il suggère une méthode qui permet de guider les projets des étudiants de façon à ce que les objectifs du programme soient atteints, tout en laissant aux enseignants la liberté d'encourager l'initiative des étudiants. Le guide comprend quatre sections. La première traite de l'initiation à l'informatique et de la méthode d'enseignement appelée "Projet étudiant". La deuxième aborde la question de l'évaluation dans le programme d'initiation à l'informatique. La troisième propose le contrat comme modèle de planification pour l'évalustion. Enfin, la quatrième partie donne des exemples de contrats que l'étudiant peut négocier avec le professeur.

Type de document: d

EDUQ
8879786

INTRODUCTION TO COMPUTER SCIENCE ICS 611-434, 611-534 EVALUATION

Direction générale des programmes
Direction de la formation générale

Translation by:
Howard Scott, with the assistance of other members of the Education Development in the English Language translation team, ministère de l'Éducation, 1988

© Gouvernement du Québec
Ministère de l'Éducation, 1988

ISBN 2-550-14438-4

Legal Deposit - Third Quarter 1988
Bibliothèque nationale du Québec

INTRODUCTION TO COMPUTER SCIENCE

ICS 611-434, 611-534 - EVALUATION

This document is an adaptation of a previous document prepared by the Montérégie working group on evaluation and the project method in the ICS course. The members of the group were:

Michel Arcouet
Laurent Bélisle
Jean-François Cloutier
Wasfi Daoud
Michel Labelle

They were assisted by:
Gilles Chainey
Céline Gagné
Bertrand A. Morin

TABLE OF CONTENTS

Foreword v

INTRODUCTION 1

Section I
ICS and the Project Method. 3

Section II
ICS and Educational Evaluation 9

Section III
THE CONTRACT: A Planning Model for
Educational Evaluation. 15

Section IV
EXAMPLES OF CONTRACTS 21

APPENDICES 43

Foreword

The Introduction to Computer Science program, or ICS, is an innovative program which has been successfully introduced into the school system in spite of many problems with its content and approach, and with the implementation process.

During the long process of implementing a program such as ICS, from conception to the classroom, it is necessary to provide support and assistance at critical moments to those primarily responsible for education in the schools, the teachers. The **Curriculum Guide** was the first document provided to support the program. The guide gave general suggestions on how to ensure that learning and teaching in the ICS program would not be prematurely "carved in stone," and formalized. The implementation strategy also provided for the addition of information documents on subjects which posed particular problems, subjects identified through regular evaluations of the various phases of implementation.

Information documents are being published on three themes. Each is intended to help improve the teacher-student relationship by providing educators with suggestions on particular methods and considerations. This process of support and assistance is intended to take place in a spirit of flexibility and respect. The three themes are: **The Student Project**, **Evaluation**, and **Problem Solving**. The present document is concerned with **Evaluation**, and it is hoped that it will make a contribution toward quality education.

Bertrand A. Morin
Ministère de l'Éducation

INTRODUCTION

1 - General Spirit of the Introduction to Computer Science Program

The program provides an "Introduction to Computer Science," i.e. an initiation to the subject in the same spirit as the introductions to the other sciences in secondary school. It is intended to provide students with an understanding of the basic elements of computer science. These elements can be divided into three categories: the subject matter of the science, the techniques it employs, and the methods which characterize computer science. The Computer Science program which preceded the ICS program dealt essentially with the first two categories. The ICS program is intended to provide a more balanced approach, with an emphasis on the construction of algorithms and on acquiring the intellectual tools necessary for this, rather than on programming itself. For this purpose, the program favours using teaching projects, since this method is particularly effective for the attainment of general objectives 1 and 2.

Teachers who are not very familiar with the project method may, however, have problems with supervision and evaluation.

2 - The Role of This Supplement to the Curriculum Guide

This guide suggests a method of guiding projects which ensures that objectives are attained, and at the same time, permits teachers to encourage student initiative. The suggested method is certainly not the only possible solution, but teachers may wish to adapt it in order to

give their students more latitude. Teachers should remember that, while supervision is important to ensure that the objectives are attained and that evaluation is equitable, if it is too strict it can change projects into mere problem-solving exercises, not only restricting student initiative, which is the main advantage of the project method, but also limiting the heuristic aspects of learning.

I

ICS and the Project Method

1 - The Project: More Than Problem Solving

First of all, it is necessary to distinguish between an exercise, a problem and a project. An exercise is an assignment given to the student. It deals with a limited and clearly defined set of concepts and skills. There are a limited number of solutions. The teacher normally assigns exercises immediately after teaching the related concepts or skills.

A problem deals with a broader set of concepts and skills. The possible solutions may be many, varied, and sometimes unforeseen. A problem may call for knowledge and organization of knowledge which has not been explicitly presented to the students. Problems are usually assigned by the teacher.

A project is a broad problem which an individual student chooses for himself/herself. Solutions are many and varied. Each project requires the use of a set of particular skills and concepts. Students may discover what skills and concepts are required only in the course of the project, and this can add to the excitement of learning. When a project is assigned rather than chosen by the student, it can be perceived as nothing more than a problem, and not give rise to the same sense of involvement on the part of the student.

2 - Project Method

With the project method, more emphasis is placed on the student as primarily responsible for his/her own learning. The student takes responsibility for the planning of his/her project, both in terms of organization and scheduling.

Individual differences become more apparent with the project approach, and require ongoing, close attention from the teacher. The project topics and the approaches used vary from student to student. Teaching and learning become an integrated activity in which the development of skills, the acquisition of knowledge, and production take place simultaneously.

Each student project is different, making norm-referenced evaluation more tricky. The order and rate of learning also vary from one student to another. At the end of the year, however, the learning accomplishments of all the students in a class should be comparable, if not identical.

The project method also has certain implications for the role of the teacher:

- The teacher supervises the choice of individual projects, becoming, in effect, a "director of a research and development centre";

- The teacher supervises the work on the projects as they develop (individual follow-up and specific help for each problem encountered by individual students), acting as a "computer consultant";

- The teacher measures the effo[illegible]ade by the students, their cognitive learning [illegible]oblem-solving skills, thus retaining his/her [illegible]tional role as the evaluator of learning.

3 - Supervision and Evaluation

The supervision of student projects requires that the teacher:

- foresee problems inherent in the choice of a project;

- evaluate the appropriateness of the choice of project in terms of the individual student's needs and abilities, and the program objectives the student must attain;

- follow the student's progress in order to help him/her complete the project;

- identify the problems encountered by the student in order to suggest possible approaches, strategies and sources of information.

The evaluation of student projects requires that the teacher:

- identify the strategies, techniques and concepts used in a program or a document;

- assess the degree to which students improve their use of these strategies, techniques and concepts.

In keeping with the program, this document suggests a model for planning and evaluation that permits the supervision and follow-up of the students' projects by the teacher, and provides a more realistic approach to evaluation. This model also respects the essential

characteristics of the project method and is designed to maintain quality evaluation.

II

ICS and Educational Evaluation

1 - The Functions of Evaluation

The "Introduction to Computer Science" program suggests five basic functions of educational evaluation:

1. making a diagnosis;
2. modifying a program or a curriculum;
3. comparing;
4. defining needs;
5. determining whether objectives (and goals) have been attained.

1. **Making a diagnosis** usually involves detecting and identifying a student's learning difficulties with a view to applying corrective teaching measures. This is one of the functions of formative evaluation.

2. **Modifying a program** encompasses both the development of the program and its implementation in the classroom.

3. **Comparing** programs, teaching methods, school organization, and other aspects of the educational system permits the evaluation of educational choices.

4. **Defining needs** involves determining the timeliness and appropriateness of actions.

5. **Determining whether objectives have been attained** is the basis of the evaluation of students' learning and the certification (recognition) which follows.

While teachers are obviously concerned with all the functions of evaluation, this guide will concentrate on those aspects that involve making a judgement on students' learning, either for formative purposes or for certification in the sense of recognizing or confirming student learning.

2 - Objectives

The orientations, objectives, methods of evaluation and other elements define a program as a whole. It is clear, however, that the goals and objectives are the key to any program.

There are four levels of objectives: global, general, terminal, and intermediate.

Global objectives form the bridge between the goals of the program and the aims of education as defined by the ministère de l'Éducation.

General objectives ensure the coherence of the program by defining broad themes. They deal with the various elements which make up computer science and with how computer science relates to the students. The first two general objectives are concerned with scientific method, the next two with techniques and materials, and the last two with the new relationship established among the student, society and the discipline (computer science).

Terminal objectives add detail to the general objectives. They describe the behaviours expected of the students. It is these behaviours which are subject to evaluation. Evaluation is not solely concerned with certifying attainment of objectives. In fact, it would

not be appropriate to give a poor mark to a student who does not attain socio-affective objectives. In this case, a critical evaluation of teaching methods and strategies should be made.

Finally, intermediate objectives specify in greater detail the terminal objectives. They relate closely to the materials and the programming languages used. Evaluating intermediate objectives is like looking at a landscape with a microscope.

3 - Conclusion

When the time comes to mark students, it is the terminal objectives which should guide the teacher. Socio-affective objectives should not be taken into account in assigning marks.

It is not necessary to measure all terminal objectives directly. Indirect measurement instruments often provide equally good information, and a more global view which takes human aspects into account.

It is hoped that the contract model developed below will meet the requirements of the program while providing the flexibility necessary for the project method.

III

THE CONTRACT

A Planning Model for Educational Evaluation

1 - Definition

Contract is used here to mean the agreement by which the teacher and student specify the requirements, methods and elements which make up a project to be carried out within the framework of the ICS program.

For the teacher, the contract is intended to ensure that the objectives of the program are attained, and that evaluation is carried out according to the principles of norm-referenced, criterion-referenced, summative, and formative evaluation. The contract also sets deadlines and defines the domain for the project.

For the student, the contract stipulates the deadlines, indicates the subject of evaluation, and specifies the times and types of evaluation. Normally, all students are given the same type of contract.

The contract is therefore a set of requirements designed to have students deal with the types of problems for which they must use a clearly defined set of skills and concepts (in accordance with the terminal objectives of the program). The contract permits a great deal of individual choice with regard to the topic of study. A progressive sequence of learning takes place through a series of contracts which make up the school year.

The teacher introduces new concepts and strategies, and illustrates their use as part of a larger project that he/she carries out for or with the class. This is called a formative project.

The evaluation of learning is based on the measurement of how well a student satisfies requirements in carrying

out the contract. It should be noted that evaluation is not directly concerned with terminal objectives. If the satisfaction of requirements ensures that the terminal objectives are attained, it is enough to evaluate whether these requirements have been met. The problem then is to clearly establish the requirements.

2 - The Basis of the Contract: Planning

The validity of the contract is based on methodical preparation which takes place in three phases: instructional planning, formulating the contract, and evaluation planning.

2.1 **Instructional Planning** consists of precisely determining:

- learning outcomes (terminal objectives): the skills to be applied and the concepts to be developed;
- related skills and concepts: the concepts, skills, prerequisites and corequisites;
- the teacher's involvement: the step in the formative project at which the teacher introduces and illustrates skills and concepts required for or related to the students' projects.

2.2 **Drawing Up the Contract** consists of:

- defining the steps: specify the steps the student must go through in carrying out the project;
- defining the requirements: for each step, specify the requirements which will ensure the application of the skills and concepts which the student will learn.

It should be emphasized that the educational validity of a contract (and the contract model itself) is based on a precise correspondence between the requirements of the contract and the terminal objectives of the program.

2.3 Evaluation Planning consists of:

- determining the measurable elements of each requirement;
- coupling these elements with a descriptive scale, for example, the requirement has been satisfied:

not at all satisfactorily completely

- giving each scale a relative weight: the teacher makes a judgement here as to the relative importance of each requirement;
- giving a relative weight to the contract: the more complicated the contract, the greater the weight it should be given; the final contract should be seen as a synthesis activity since it is the most important (in fact, it replaces the traditional final examination).

IV

EXAMPLES OF CONTRACTS

1 - Preliminary Notes

The schedule suggested by the ministère de l'Éducation for the ICS program (4 classes or periods of 50 minutes over 6 days) has been used to set the durations of the contracts.

The examples can be applied directly to LOGO, Pascal and some versions of APL. For BASIC, it will be necessary to simulate procedures through well structured programming. The contract concept can therefore be applied to any programming language (including others not mentioned above).

Each contract is interspersed with ad hoc teaching. This teaching takes the same amount of time as the contract itself. A 5-week contract means that 5 weeks of work on the contract are distributed over 10 weeks. In order to familiarize the students with the kind of planning and analysis involved in the project, the teacher carries out a so-called formative contract. The series of contracts is preceded by a brief introduction to the computer, to simple languages and to program writing.

2 - Before Starting the First Contract

The following is a brief example of the activities that must be carried out before starting the first contract:

- operating the computer;
- primitives: a few graphics and writing commands;
- elementary repetitive and recursive structures and commands:
  - REPEAT (repeat structure);
  - creating procedures;
- using the editor (working knowledge);
- managing the work space:
  - saving and retrieving, listing the names of procedures;
  - deleting procedures;
- defining commands to draw regular shapes, first without arguments, then with arguments;
- concepts of iteration and recursion with a queue;
- beginning the teaching project (formative project), for example, the game of tic-tac-toe.

DURATION: 8 to 10 classes.

Note: This short duration indicates that this project is an introduction. Topics are not dealt with in depth.

3 - First Contract

Topic: To create a procedure to draw a complex shape.

3.1 Instructional Planning

Anticipated outcomes:

- to construct simple algorithms;
- to group and name actions;
- to use meaningful names;
- to order actions;
- to use primitives and procedures.

The objectives of the program which are particularly emphasized:

1.1-1.3, 1.5-1.9;
2.1-2.6, 2.9, 2.11, 2.13-2.14;
3.1, 3.2, 3.5, 3.6, 3.9-3.11, 3.13, 3.14.

The following questions should be asked:

- what is the status (e.g. states of a turtle: position, direction, pen)?
- how should the actions of defined commands be ordered?
- what is the relation between a procedure and a sub-procedure?

Teaching strategies:

The formative project that is carried out in class for the purpose of showing the students how to do a project is tailored to the wishes of the students. Procedures and variables, if applicable, are given meaningful names. The concept of state should be used, and the problem of the ordering and the structuring of proce-

dures should be clearly presented. Teaching sessions should by synchronized with the implementation of the various steps of the project.

3.2 Drawing Up the Contract

Steps:

- choice of topic and outline (1 class);
- project structure (1 class);
- program (4 classes).

Requirements:

- there are at least three types of regular polygons;
- the shape contains at least 6 elements;
- the procedures and arguments have meaningful names.

3.3 Evaluation: Elements evaluated and weighting (out of 100)

- structure (50):
  - meaningful names (15);
  - at least 3 polygons (15);
  - at least 6 elements (20);
- program (50):
  - consistency with structure (25);
  - it works! (25).

All documents MUST be handed in.

DURATION: 6 classes over 4 weeks. Other classes over and above the 6 allocated for the contracts are used for the formative project.

4 - Second Contract

Topic: To write a program that draws a complex object, taking its structure into account

4.1 Instructional Planning

Anticipated outcomes:

- to make a structured top-down analysis;
- to divide a problem into sub-problems (into polygons);
- to generalize (by using arguments);
- to make analogies (isomorphisms);

The objectives of the program which are particularly emphasized:

1.1-1.9, 2.7, 2.8, 3.3, 3.4, 3.7.

The following questions should be asked:

- how are arguments used to generalize a procedure?
- what is a hierarchy, a structure?
- how is a structured top-down analysis done?

Teaching strategies:

This second project is much broader in scope than the first. Its purpose is to help the student to better structure his/her work and to make generalizations.

The teacher assists students in structuring programs by encouraging them to observe the structures of the object to be drawn and by suggesting they write procedures for each of the principal parts of the object.

Sub-procedures correspond to each element of the principal parts. Refinement takes place, in descending order, both of the analysis of the object and the analysis of the program being written. From the point of view of program efficiency, isomorphism with the object is not always the best method. However, it often is the most efficient from the point of view of maintaining and developing the software. Since skills and knowledge related to information processing are the most important elements in the ICS program, the approach described above is likely to be the most useful.

Skills in top-down analysis are basic to structured programming. This is why they should be introduced as soon as possible. If students carry out these activities early in the year when knowledge of programming is still rudimentary, they will consider these methods effective, and subsequently, will use them spontaneously instead of improvising.

Generalisation is accomplished through the systematic use of arguments for procedures. Students should be made aware that the most general procedures are often procedures with several arguments. In addition, they are often shorter and simpler than procedures which are dedicated solely to one task.

Finally, the concept of state should be emphasized: the state of variables, the state of the turtle, and so on. Within each procedure, two steps can be distinguished: the preparation of states (initialization of the position of the turtle, initialization of the variables) and the work to be performed. As soon as it becomes a little complicated, the latter step is assigned to a sub-procedure.

Since it involves the same type of activities as those expected from the students, the formative project is the basic vehicle for teaching the activities, methods and concepts mentioned above.

4.2 Drawing Up the Contract

Steps:

- choice of topic and outline (1 class);
- structural analysis (2 classes);
- programming (6 classes).

Requirements:

- there are at least 10 elements;
- the hierarchy contains at least 4 levels;
- congruity of the hierarchical structure with the structure of the program;
- each procedure (except for procedures to draw single polygons) contains only initializations and sub-procedures;
- use of arguments or parameters.

4.3 Evaluation: Elements evaluated and weighting (out of 100)

- choice and outline (10);
- precision of grid (10);
- structural analysis (50):
  - at least 10 elements (10);
  - hierarchy (40);
    - strict, with nodes named (30);
    - sheets = polygons (10);
- programming (30):
  - structural congruity (10);

- procedures: initialization followed by a sub-procedure (10);
- it works! (10).

All documents MUST be handed in.

DURATION: 9 classes over 6 weeks. The other classes are allocated to the formative project and other forms of teaching.

5 - Third Contract

Topic: To write a dialogue consultation program based on a decision tree.

5.1 Instructional Planning

Anticipated outcomes:

- to classify;
- to iterate:
  - as a heuristic exercise: repeated use of a structure;
  - as a processing structure;
- to model (in this instance a dialogue);
- to collect information: to derive content;
- to use top-down analysis;
- to do structural and analytical definition (decision tree);
- to construct alphanumeric expressions;
- to control display and input.

The objectives of the program which are particularly emphasized:

the objectives of the preceding contract, with special emphasis on 1.1-1.9, 2.1-2.7, 2.11-2.13, 3.1-3.9, 3.11, 3.13.

The following questions should be asked:

- how are interactive primitives used?
- what is a simple function?
- how does one work with words and lists?
- how and when are structures such as CHOICE, IF...THEN and UNTIL used?
- what is pseudo-code and how is it used?
- what is the process of deduction?

- what is a global variable and a local variable; how and when are they used?
- what are decision trees, and how and when should they be constructed?

Teaching strategies:

The theme of this third contract is less concrete than those of the first two. It is much more difficult to visualize the structure of a conversation than that of an image. The structure, however, is a simple decision tree.

This contract provides an opportunity to explore various notions of computer processing, such as "CHOICE" structures and various types of repetition. These are described using pseudo-code. Pseudo-code should be flexible. It should never be as rigid as a programming language. It is not "Pascal" in plain English, but rather a descriptive tool which creates a bridge between everyday language and programming language.

For this first approach to data structures, the work will be limited to simple structures: words, lists or chains. Here again, the emphasis will be on planning a top-down analysis. The deduction process will also be an important element.

In order to make the notions dealt with in this contract concrete and practical for the students, it is important to include an application of this type in the formative project. In this way, computing techniques are introduced in immediate relation to their applications.

Choosing the steps for carrying out the contract is not an insignificant task. Before the pseudo-code is written, a large part of the contract is devoted to the written verbal description of steps, and to the description of the structure. In fact, descriptive skills are more fundamental than very specific techniques related to pseudo-code. These are the skills which are more likely to be applicable (or transferable) to other disciplines. In any case, students have to know what they want to do in order to be able to write pseudo-code. One way to acquire this ability is through proper analysis methods and by using many heuristics as suggested in the document which describes the ICS program.

5.2 Drawing Up the Contract

Steps:

- choice of domain (1 class);
- description in English of the project (2 classes);
- description of the structure of the tree and the procedures: for each node give the name and the behaviour (3 classes);
- pseudo-code (3 classes);
- coding (7 classes).

Requirements:

- 3 steps in the program: introduction, consultation and results;
- the program asks questions, then records, checks, and interprets the answers;
- the interaction is personalized;
- each node in the tree corresponds to a procedure in the program;

- certain nodes in the tree have more than 2 branches.

5.3 Evaluation: Elements evaluated and weighting (out of 100)

- verbal description (25);
  - domain well analyzed and defined (10);
- tree (25):
  - mutual exclusion of branches (10);
  - correspondence between nodes and procedures (10);
  - at least one node has more than 2 branches (5);
- pseudo-code (25):
  - isomorphism with the tree (6);
  - correct use of control structures (7);
  - 3 phases of the program (6);
  - question, reading, checking and interpretation; all present (6);
- coding (25);
  - faithful to the pseudo-code (8);
  - proper implementation of control structures (9);
  - it works! (8).

All documents MUST be handed in.

DURATION: 16 classes over 8 weeks. The other classes are allocated to the formative project and other forms of teaching.

6 - Fourth Contract

Topic: Student's choice (for this contract, a preplanned example is presented as an illustration).

6.1 Instructional Planning

Anticipated outcomes:

- the same elements as 5.1 plus:
  - ▸ structured, hierarchical top-down analysis;
  - ▸ breakdown of a very complex problem into several parts;
  - ▸ development of a program including a large set of interrelated and interdependent parts;
  - ▸ a presentation of the project to the other students in the class.

The objectives of the program which are particularly emphasized:

- the objectives of the preceding contracts;
- special emphasis on 1.4, 1.8, 1.9, 2.5, 2.6, 2.11, 2.14-2.20, 3.1-3.15, 4.3-4.6, 6.1-6.5.

The following should be taken into account:

- emphasis on work method: readability of program and clear identification of subprograms, procedures and variables;
- an opportunity to apply new concepts immediately;
- the new concepts are, among others (in the example):
  - a. to introduce a set of elements in a vector;
  - b. to verify if an element is part of a vector:

c. to choose and derive one or more elements from a vector;

d. to add elements to a vector;

- construction of the hierarchical structure for a project;
- determination and implementation of the various components which make up the program;
- use of the main structures of the program;
- use of varied data structures;

Teaching strategies:

The following contract project is an example of an end-of-year synthesis project. This fourth project is important because of its scope and the variety of its elements. Both the analysis carried out before programming and the construction of the whole program will require special attention. The students may carry out the project individually or in groups, but it may be useful to plan to "correct" group projects in order to ensure that each student benefits equally.

Good planning, well defined strategy, careful programming, and complete documentation will keep improvisation to a minimum, while at the same time allowing the students to explore computer programming fully and naturally. The projects should be announced several weeks ahead of time, and students should be permitted to chose their own project topics.

At some stage, students should be asked to justify and explain their approach. This will facilitate evaluation.

The example:

The construction of a crossword puzzle limited to a horizontal mode of operation, a limitation which has been deliberately set. The steps of the project are essentially the following:

- to construct a 10 X 10 grid;
- to choose words that will fit in the grid;
- to find clues that correspond to the words;
- to locate and place white and black squares in the grill;
- to insert the words in the appropriate places in the grid;
- to validate the crossword puzzle.

The description of the game is relatively demanding. It should include an outline that is sufficiently clear that any player can do the puzzle without any further explanation, and at the same time, is not too long. The outline should include the rules of the game and a brief explanation. This part of the exercise has as much to do with language skills as it does with computer science.

Once the game has been described, the programming portion of the exercise must be outlined. The operations that the computer has to perform will be specified in cycles. First, a list of these operations is made so that the whole process can be clearly understood. Then, precise instructions for the main program and all the subprograms at the various levels are written. These two operations must be repeated until an adequate result is obtained. A bottom-up analysis process is used here. Alternating between subprograms and the whole program may also be useful, or even essential, here. The step

of numbering and naming the branches and hierarchical levels and their constituent parts is particularly important, and both teacher and student should pay special attention to this phase.

The "rough" programming then begins with the main program and continues with the various components of the main program. During this step, the student should demonstrate logical programming and work methods, and respect for basic programming rules which should have already been learned. Notions studied earlier may be applied to this year-end project.

This is the time to introduce new analysis techniques and new knowledge of the programming language used, and to apply important concepts and synthesis methods. The paragraph on what "should be taken into account" in each contract example deals with this aspect.

It is important to alternate between theory and practice. This will help to prevent many errors in judgement and negative feelings concerning the relevance of the knowledge acquired. Theoretical knowledge and the applications of this knowledge are complementary. The following are examples.

T. How to read data (in Read Data in BASIC) and "fill" a vector with this data.

A. Enter clues, answers, positions (subprograms such as 11..., 12..., and 13...).

T. Return to the three types of repetitive structures:
   - repeat until,
   - repeat,
   - while.

A. Write the subprogram (22 000):
   repeat until (30 times) or until (the crossword is completed).

T. How to add elements to a vector while the program is being run.

A. Write the subprogram (50 000):
   add the solution to the vector.

T. How to check if an element belongs to a vector.

A. Write the subprogram (40 000):
   the computer checks if an answer has been given for the clue.

Supplementary Applications: the computer displays the answers directly on the grid using the elements in the answer vectors and position vectors.

Finally, it should be added that even a complete, clearly defined hierarchical structure is not closed. On the contrary, it can, by definition, be constantly improved. The student can add music and graphics in order to refine his/her software product. The worst possible thing that could occur would be for the student to be placed in a work situation which would lead to a dead end because he/she is unable to correct or create his/her individual project. The same is true if the working materials --- computers, hardware, software --- available to the student are so limited that he/she cannot complete the project as desired. Emphasis on

individualized school work is an enriching educational approach which fosters good student-teacher relations.

6.2 Drawing Up the Contract

Steps:

- introduce the project;
- make the student aware of the work which the computer will have to do to solve the problem (1 class);
- describe the project in detail (3 classes);
- describe the steps leading to programming (1 class);
- organize the steps in the form of a hierarchical structure (2 classes);
- number the hierarchical structure (1 class);
- give the program and the subprograms meaningful names; also name variables and other elements (2 classes);
- begin the process of programming, respecting the structure and plan established (13 classes);
- evaluate the project using a questionnaire (1 class).

Requirements:

- the project should be original;
- basic instructions must be followed, particularly with regard to student autonomy in carrying out the project;
- it is essential that all documents that are part of the planning, including support material for the project, be handed in;
- clear identification of program structures is important;

- three key steps in the program are the introduction, the consultation, and the results;
- the program uses a two-option menu and provides the information necessary (consider possible enrichment);
- the program includes a single input and a single output;
- the program asks questions, notes answers, checks them and interprets them;
- the program includes at least three levels and at least ten procedures.

In total, a project of this type requires about 24 hours of intensive classroom work, or about 2 months.

6.3 Evaluation: Elements evaluated and weighting (out of 100)

Since this is a contract for the end-of-year project, students should be marked using an approach which permits the teacher to remain open and flexible. Even though the school report will include comments on all documents handed in, the number mark is based on the program and the overall project.

- global structure of the project (30):
  - ▸ description of tasks;
  - ▸ hierarchical structure;
  - ▸ identification of variables and procedures;
- program (50):
  - ▸ presentation;
  - ▸ structure of the program:
    - · use of the menu;
    - · proper use of procedures;
    - · use of appropriate program structures;
  - ▸ the program runs correctly;

- the program is readable;

- quality of presentation of the whole project (20):
  - description;
  - originality;
  - independence demonstrated in carrying out the project.

All documents MUST be handed in.

These are minimum requirements that indicate what the student must accomplish in order to obtain a passing mark of 60%.

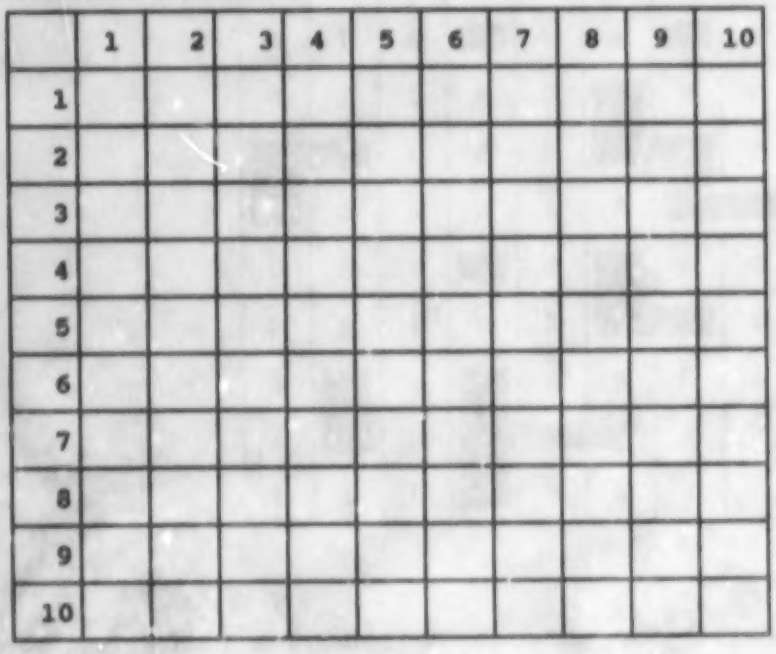
NOTE:

The example of an end-of-year contract is accompanied by appendices which illustrate the component parts. This is why the description looks a little different. These appendices, as well as the information contained in the earlier sections, should be kept in mind when reading Contract 4.

This sample project also provides a link with the information document dealing specifically with the student's project.

APPENDICES

The following is an example of the crossword puzzle grid at various times in its development, and the accompanying definitions and solutions.



Horizontally

1 a) action of a celestial body in an orbit

2 a) refusal

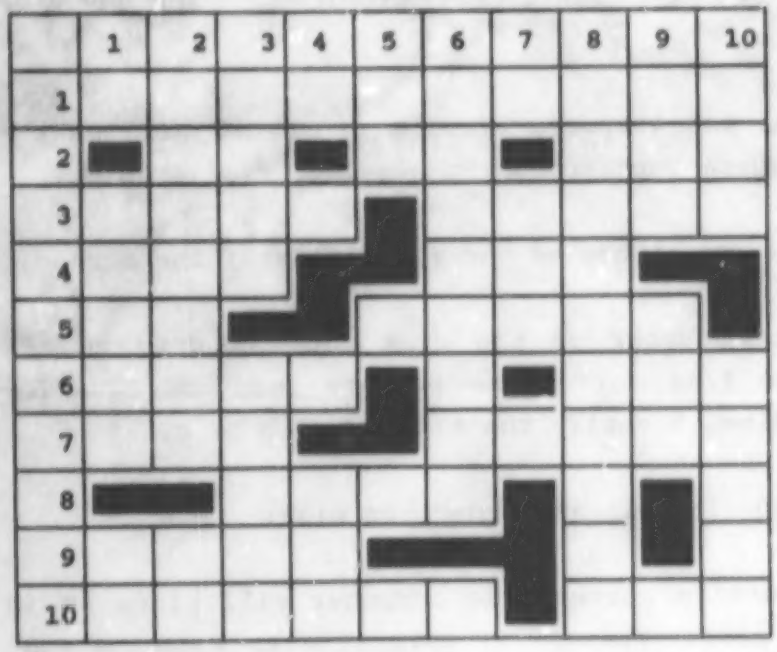
b) article

c) Italian possessive

3 a) helmet stuff

b) Zola

.....



Vertically

1 a) joyous songs

b) pronoun

2 a) brain-teasers

b) sales pitch

3 a) what some do with feet

b) what many do on feet

.....

	1	2	3	4	5	6	7	8	9	10
1	R	E	V	O	L	U	T	I	O	N
2		N	O		A	N		M	I	A
3	P	I	T	H		E	M	I	L	E
4	A	G	E			R	A	T		
5	E	M			B	A	N	A	L	
6	N	A	S	A		S		T	A	U
7	S	S	T			E	D	I	T	S
8			A	X	E	D		O		E
9	M	A	N	X				N		R
10	E	D	D	I	E	S		S	I	S

· The object of the game is to find the words for the grid from the clues (definitions) provided. You have 30 tries.

· To begin, the computer draws a crossword grid on the screen. Each square corresponds to one letter of a word.

· All clues are listed on the right side of the screen.

· Choose the number of the clue corresponding to the word you wish to find. If there is more than one clue for one horizontal line, identify the clue as a, b or c.

· Answer the clues in any order you wish.

· If the word is correct the computer will place it in the grid in the proper place. If not, it will indicate your error.

• Continue the game until you have found all the words or used up your 30 tries.

• If you succeed in completing the crossword puzzle, the computer declares you the winner.

GOOD LUCK!

- A menu which gives a choice of:
    1. explanation,
    2. crossword puzzle,
    3. exit.

- If 2. is chosen, the computer draws the grid and adds the filled-in or black squares.

- The computer takes the clues from its bank of clues and displays them to the right of the grid.

- The correct answers are inserted into the program in the form of data. The computer enters or places the answers or elements in a vector.

- The horizontal positions are inserted into the program in the form of data. The computer enters these elements in a vector.

- The player chooses a clue number. The computer checks whether the clue number has already been answered by consulting its bank of answered clue numbers. If the clue has already been answered, the player is asked to make another choice.

- The computer waits for a response. The player enters his/her answer.

- The computer evaluates the response given by comparing it to the one that corresponds to the chosen clue in its bank of answers.

- If the answer is correct, the computer consults its bank of positions to find the exact location where it should begin writing the word in the grid. The computer keeps

track of the correct answers and adds this clue number to the bank of answered clues.

· If the answer is incorrect, the computer indicates the error.

· The player may ask for the clue again if he/she did not find the correct answer.

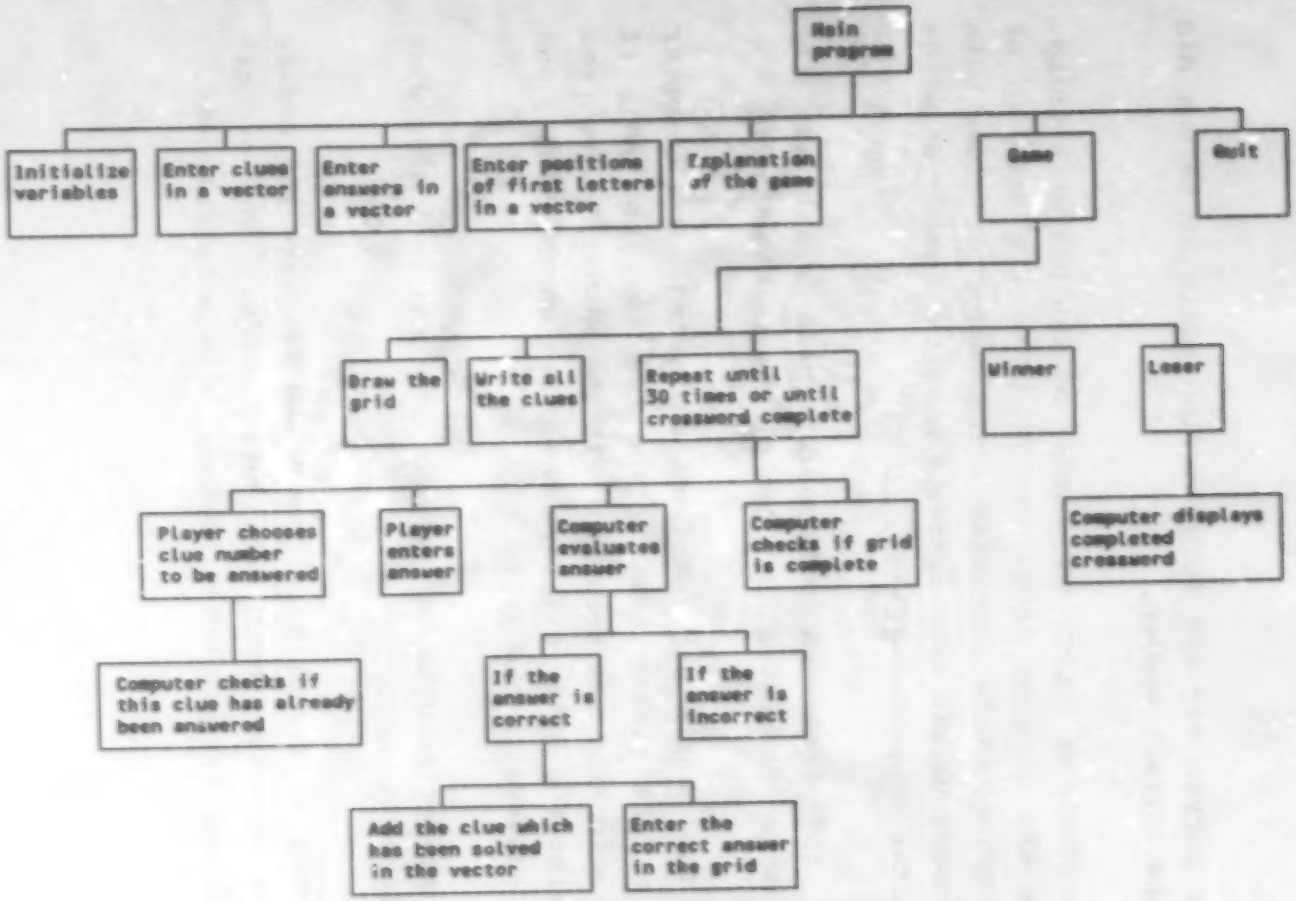
· The computer counts the number of the player's tries. It repeats the process until the 30 tries are used up or until the crossword is completed. The number of clues with correct answers should then correspond to the number of words entered in the crossword grid.

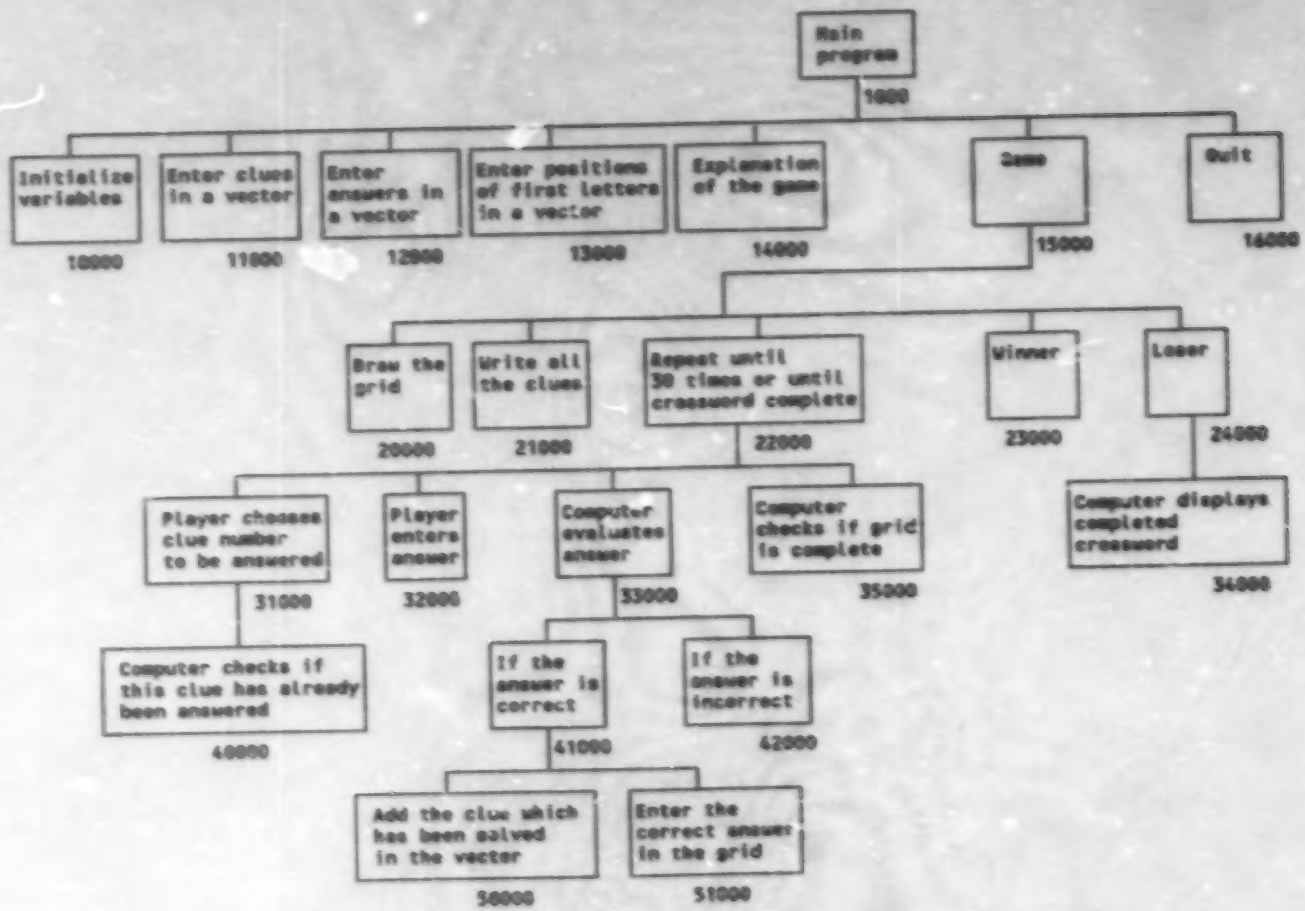
· If the crossword puzzle has been completed, the computer display indicates that the player is the winner.

· If the puzzle has not been completed, the computer indicates to the player that he/she has lost and asks if he/she wants to know the solution. If the answer is yes, the computer draws the grid, displays the clues on the right, and places the answers in the grid.

· The player returns to the main menu and indicates what he/she wants to do next.

N.B. This is a working document and teachers may modify parts of it to suit their particular needs. The overall process outlined for projects, however, should be followed.


Main program
Initialize variables
Enter clues in a vector
Enter answers in a vector
Enter positions of first letters in a vector
Explanation of the game
Game
Quit
Draw the grid
Write all the clues
Repeat until 30 times or until crossword complete
Winner
Loser
Player chooses clue number to be answered
Player enters answer
Computer evaluates answer
Computer checks if grid is complete
Computer displays completed crossword
Computer checks if this clue has already been answered
If the answer is correct
If the answer is incorrect
Add the clue which has been solved in the vector
Enter the correct answer in the grid


Main program
1000
Initialize variables
10000
Enter clues in a vector
11000
Enter answers in a vector
12000
Enter positions of first letters in a vector
13000
Explanation of the game
14000
Game
15000
Quit
16000
Draw the grid
20000
Write all the clues
21000
Repeat until 30 times or until crossword complete
22000
Winner
23000
Loser
24000
Computer displays completed crossword
34000
Player chooses clue number to be answered
31000
Player enters answer
32000
Computer evaluates answer
33000
Computer checks if grid is complete
35000
Computer checks if this clue has already been answered
40000
If the answer is correct
41000
If the answer is incorrect
42000
Add the clue which has been solved in the vector
50000
Enter the correct answer in the grid
51000

Gouvernement du Québec
Ministère
de l'Éducation

16-0081-04A
[illegible]